

《C/C++ 学习 指南》

第21.3讲：构造与析构

作者：邵发 QQ群：417024631
官网：<http://www.afanihao.cn/>
答疑：<http://www.afanihao.cn/kbase/>

C/C++学习指南 邵发 www.afanihao.cn

内 容 提 要

1. 默认构造函数
2. 成员的构造与析构顺序
3. 成员的初始化列表

默认构造函数

把那种不需要传参的构造函数，称为默认构造函数。

例如，

```
Object();
```

或者构造函数的所有参数都有缺省值，也称为默认构造函数

```
Object(int a=10, int b=11);
```

(函数默认值的语法参考第8章)

有了默认构造函数之后，对象在构造时就可以不传递参数。例如，
`Object obj;`

默认构造函数

默认构造函数很重要。如果一个类没有默认构造函数，则无法构造数组。

例如，

```
class Object
{
public:
    Object(int x) {}
};
```

```
Object objs[4]; // 错误!
```

所以，一般都要给类定义一个默认构造参数，以方便使用。

没写构造函数？没写析构函数？

如果一个类没有写任何构造函数，则编译器隐含地生成为其添加一个，相当于

// 相当于添加了

```
Object::Object()
```

```
{  
}
```

如果没写析构函数，则编译器隐含地生成一个析构函数相当于添加了

```
Object::~~Object()
```

```
{  
}
```

（注意：只有当你没有写构造函数的时候，才给隐含地加一个构造函数。一旦你写了一个构造函数，它就不给你加了）

成员的初始化与析构

考虑成员变量本身也是 class 类型的情况：

1. 当对象被构造时，成员变量也被构造（成员变量的构造函数被调用）
2. 当对象被析构时，成员变量也被析构（成员变量的析构函数被调用）

成员的初始化与析构

结论：

一、构造的时候

- (1) 成员被依次构造：从前到后
- (2) 先执行成员的构造函数，再执行自己的构造函数

二、析构的时候

- (1) 成员被依次析构：从后到前
- (2) 先执行自己的析构函数，再执行成员的析构

初始化列表

可以在构造函数后面直接初始化

- (1) 以冒号引导
- (2) 使用小括号来初始化

初始化列表

(1) 基本类型的初始化

```
class Object
{
public:
    Object() : x(1), y(2)
    {
    }
};
```

初始化列表

(2) class类型的初始化

用小括号带上构造参数

```
class Object
{
public:
    Object() : m_child(1,2)
    {
    }
};
```

初始化列表

“初始化”与“赋值”有何区别？

(1) 功能上：一样

(2) 性能上：“初始化”相对较好，有时可以节省CPU操作

```
class Object
{
public:
    Object() // 已经对 m_child进行了构造
    {
        m_child.x = 1;
        m_child.y = 2; // 又多了一步操作
    }
};
```

小结

1. 默认构造函数：不需要传参的构造函数

这个非常必要，一般的类都要提供一下默认构造函数

2. 构造与析构的顺序问题

3. 初始化列表

在构造函数的位置指定成员的初始化构造参数